CLAIMS

We claim:

1. A method of distributed collaborative computing comprising:

     partitioning a collaboration function into sub-functions;

     assigning at least one said sub-function to each of a plurality of logical processes;

     associating a respective management process with each of said plurality of logical processes, said logical processes configured so that each said logical process is capable of communicating with every other said logical process thru said respective management process;

     communicating between said logical processes using said respective management processes; and

     monitoring said respective management processes with a single supervisor process;

wherein said monitoring further comprises re-creating one or more said logical processes in response to detecting a failure of one or more said logical processes.

2. The method of Claim 1, said detecting further comprising:

     monitoring a message stream to determine a responsiveness of said logical process; and

     if said responsiveness ceases, signaling said failure to said single supervisor process.

3.　　The method of Claim 1, said re-creating
further comprising:
　　spawning a new logical process;
　　assigning to said new logical process said at
5　　　　least one sub-function corresponding to said
　　　　failed logical process;
　　recovering the state of said failed logical
　　　　process into said new logical process; and
　　associating a new management process with said new
10　　　　logical process.

4.　　The method of Claim 1, said re-creating
further comprising:
　　activating a stand-by logical process, said stand-
　　　　by process having been instantiated prior to
15　　　　said detecting;
　　assigning to said stand-by logical process said at
　　　　least one sub-function corresponding to said
　　　　failed logical process;
　　recovering the state of said failed logical
20　　　　process into said new logical process;
　　associating a new management process with said
　　　　stand-by logical process; and
　　spawning a new stand-by logical process.

25

5.　　A computer program for use in distributed
collaborative computing, comprising computer
instructions for:
　　partitioning a collaboration function into sub-
30　　　　functions;

assigning at least one said sub-function to each
of a plurality of logical processes;
associating a respective management process with
each of said plurality of logical processes,
said logical processes configured so that
each said logical process is capable of
communicating with every other said logical
process thru said respective management
process;
communicating between said logical processes using
said respective management processes; and
monitoring said respective management processes
with a single supervisor process;
wherein said monitoring further comprises re-creating
one or more said logical processes in response to
detecting a failure of one or more said logical
processes.

6.   The computer program of Claim 5, said
detecting further comprising:
monitoring a message stream to determine a
responsiveness of said logical process; and
if said responsiveness ceases, signaling said
failure to said single supervisor process.

7.   The computer program of Claim 5, said re-
creating further comprising:
spawning a new logical process;
assigning to said new logical process said at
least one sub-function corresponding to said
failed logical process;

recovering the state of said failed logical
process into said new logical process; and
associating a new management process with said new
logical process.

5   8.   The computer program of Claim 5, said re-
creating further comprising:
activating a stand-by logical process, said stand-
by process having been instantiated prior to
said detecting;
10   assigning to said stand-by logical process said at
least one sub-function corresponding to said
failed logical process;
recovering the state of said failed logical
process into said new logical process;
15   associating a new management process with said
stand-by logical process; and
spawning a new stand-by logical process.

20   9.   A computer-readable medium storing a computer
program executable by a plurality of server computers,
the computer program comprising computer instructions
for:
partitioning a collaboration function into sub-
25   functions;
assigning at least one said sub-function to each
of a plurality of logical processes;
associating a respective management process with
each of said plurality of logical processes,
30   said logical processes configured so that

each said logical process is capable of
communicating with every other said logical
process thru said respective management
process;

5   communicating between said logical processes using
said respective management processes; and
monitoring said respective management processes
with a single supervisor process;
wherein said monitoring further comprises re-creating

10   one or more said logical processes in response to
detecting a failure of one or more said logical
processes.

      10.   The computer-readable medium of Claim 9, said
detecting further comprising:

15   monitoring a message stream to determine a
responsiveness of said logical process; and
if said responsiveness ceases, signaling said
failure to said single supervisor process.

      11.   The computer-readable medium of Claim 9, said

20   re-creating further comprising:
spawning a new logical process;
assigning to said new logical process said at
least one sub-function corresponding to said
failed logical process;

25   recovering the state of said failed logical
process into said new logical process; and
associating a new management process with said new
logical process.

12. The computer-readable medium of Claim 9, said re-creating further comprising:

activating a stand-by logical process, said stand-by process having been instantiated prior to said detecting;

assigning to said stand-by logical process said at least one sub-function corresponding to said failed logical process;

recovering the state of said failed logical process into said new logical process;

associating a new management process with said stand-by logical process; and

spawning a new stand-by logical process.

13. A computer data signal embodied in a carrier wave, comprising computer instructions for:

partitioning a collaboration function into sub-functions;

assigning at least one said sub-function to each of a plurality of logical processes;

associating a respective management process with each of said plurality of logical processes, said logical processes configured so that each said logical process is capable of communicating with every other said logical process thru said respective management process;

communicating between said logical processes using said respective management processes; and

monitoring said respective management processes
with a single supervisor process;
wherein said monitoring further comprises re-creating
one or more said logical processes in response to
5   detecting a failure of one or more said logical
processes.

14.   The computer data signal of Claim 13, said
detecting further comprising:
monitoring a message stream to determine a
10   responsiveness of said logical process; and
if said responsiveness ceases, signaling said
failure to said single supervisor process.

15.   The computer data signal of Claim 13, said
re-creating further comprising:
15   spawning a new logical process;
assigning to said new logical process said at
least one sub-function corresponding to said
failed logical process;
recovering the state of said failed logical
20   process into said new logical process; and
associating a new management process with said new
logical process.

16.   The computer data signal of Claim 13, said
re-creating further comprising:
25   activating a stand-by logical process, said stand-
by process having been instantiated prior to
said detecting;

assigning to said stand-by logical process said at
   least one sub-function corresponding to said
   failed logical process;

recovering the state of said failed logical

5   process into said new logical process;

associating a new management process with said
   stand-by logical process; and

spawning a new stand-by logical process.